IJICTR
International Journal of Information & Communication Technology Research

Volume 10- Number 3 – Summer 2018 (31 -41)

A Hybrid DOS-Tolerant PKC-Based Key Management System for WSNs

Hamzeh Ghasemzadeh
Department of Communicative Sciences and Disorders, MSU
Department of Computational mathematics science and engineering, MSU
East Lansing, MI, USA
ghasemza@msu.edu

Ali Payandeh*
ICT Department, Malek-e-Ashtar University of Technology
Tehran, Iran
payandeh@mut.ac.ir

Mohammad Reza Aref
Information Systems and Security Lab, Department of Electrical Engineering, Sharif University of Technology
Tehran, Iran
aref@sharif.edu

Received: 14 February, 2018 - Accepted: 2 June, 2018

Abstract—**Security is a critical and vital task in wireless sensor networks (WSNs), therefore different key management systems have been proposed, many of which are based on symmetric primitives. Such systems are very energy efficient, but they lack some other desirable characteristics. On the other hand, systems based on public key cryptography (PKC) have those desirable characteristics, but they consume more energy. Recently based on authenticated messages from base station (BS) a new PKC–based key agreement protocol was proposed. We show this method is susceptible to a form of denial of service (DOS) attack where resources of network can be exhausted with bogus messages. Then, we propose two different improvements to solve this vulnerability. Simulation results show that these new protocols retain desirable characteristics of the basic method but solve its deficiencies.**

Keywords- wireless sensor network, Key Management, Broadcast Authentication, Public Key Cryptography

I. INTRODUCTION

WSNs have attracted researchers from various fields over the past decade. These specialized networks are decentralized, self-organized and can be deployed without requiring the existence of a supporting infrastructure. Basically, WSNs serve as an interface to the real world and gather some physical information from their surroundings. Thus, they have found a wide range of applications. Romer et al. surveyed many practical WSN projects [1]. Other more recent applications of WSNs include mining underground coal [2], environmental disaster monitoring [3], monitoring soccer players for injuries [4], laboratory tutoring [5], secure capturing of voice [6], military vehicle tracking [7], and many more. Unfortunately, wireless connectivity, absence of physical protection, and the unattended deployment, make WSNs prone to different types of attack. Consequently, for gathering reliable information these networks should be protected with appropriate security mechanisms, many of which rely on existence of secure keys between different nodes of the network, a task that key management system (KMS) addresses.

Recently, many KMS have been proposed for WSNs. First, a method based on the probabilistic pre-distribution of subsets of a key-pool was proposed [8]. This method had low resiliency and its connectivity was poor. Later, methods based on symmetric polynomials and generating matrices of linear codes were proposed [9, 10]. These methods solved problem of connectivity, but they had threshold resiliency and addition of new nodes to the network was hard. LEAP was designed to support secure in-network processing [11]. But if the transitory initial key of LEAP is discovered, security of the entire network is compromised. In [12] a hash-based

* Corresponding Author

mechanism was employed to enhance the resiliency of key pre-distribution schemes against node capture. It was shown that this scheme improves resiliency of q-composite method [13]. But this scheme cannot achieve perfect resiliency and it inherits other undesirable characteristics of the underlying method. Çamtepe et al. proposed a key pre-distribution scheme based on symmetric balanced incomplete block design [14]. Their scheme had good connectivity, but it did not scale very well. To improve scalability of pre-distribution-based systems, mapping from unitals to key pre-distribution was proposed [15]. But this approach does not guarantee perfect key sharing. Finally, biometric-based authentication and a two-factor authentication method based on attribute and password were proposed in [16, 17], respectively.

Another possible path to KMS is to use public key cryptography (PKC). PKC-based systems have many desirable characteristics. They provide perfect resiliency and perfect global, local, and node connectivity [18]. Furthermore, they are scalable and extensible [18]. On the other hand, their energy consumption is high. In the past decade, different PKC-based KMSs have been proposed. First, feasibility of PKC-based KMS in WSNs was investigated in [19]. Later, a more rigorous analysis was conducted in [20]. Then, TinyPK was proposed [21] a scheme which is vulnerable to the man in the middle attack [22]. Yeh et al. used Elliptic Curve Cryptography (ECC) for user authentication [23]. Security flaw of this method was detected later on [24].

To reduce energy consumption of PKC-based systems two different paths in the literature have been pursued, identity-based PKC and hybrid methods. Identity based cryptography differs from the conventional PKC in the sense that authenticity of users' public data does not need explicit verification [25]. First, Hess's identity signature scheme was employed to achieve authenticity of broadcast messages [26]. IMBAS another identity-based method reduced energy consumption of authentication [27]. Later, Rabin-Williams signature was used for authenticating the code dissemination process [28]. To further reduce energy consumption, Shim et al. proposed a pairing-optimal identity-based system with message recovery method [29]. In the second path, mechanisms based on symmetric cryptography were exploited to verify authenticity of users' public data and hence we will refer to them as hybrid approaches. First, Merkle hash tree was exploited [9]. Later, Ren et al. employed bloom filter and Merkle hash tree [30]. Another method used ECC and hash functions to authenticate broadcast messages [31]. Finally, in [32, 33] broadcast authenticated PKCs (BA) was proposed. In this method, authentication of public keys was replaced with one-time-signatures based on broadcasted messages from base station (BS). Authentication of broadcast messages of this method was governed by µTesla protocol [34]. In µTesla broadcast messages are authenticated using a message authentication code (MAC) and a key selected from a hash-chain (K*i*). To that end, the lifetime of the network is divided into a set of intervals. Then, within each interval *i*, all of broadcast messages are appended with their MAC generated using key K*i*. At the start of next interval (*i*+1), K*i* is disclosed for nodes and they use it to check authenticity of all broadcast messages they received during interval *i*. Fig. 1 presents a simple schematic of this process.

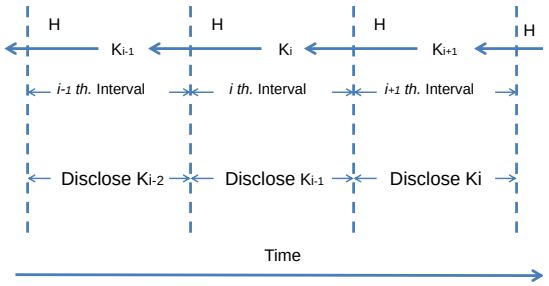


Figure 1. µTesla broadcast authentication protocol [33]

Energy is one of the main concerns in the WSNs. To improve lifetime of the network, more energy efficient mechanisms should be employed. Continuing on our seminal work [33], this paper tries to reconcile between high security demand of critical applications and energy consumption of PKC-based systems. This paper makes the following contributions:

- BA method relied on a modified version of µTesla protocol for authenticating messages from BS. Delayed nature of µTesla opens the door for denial of service (DOS) attacks. Impact of such attacks are investigated thoroughly in this paper.

- To mitigate DOS attacks, two different strategies based on hash function and bloom filter are proposed.

- Susceptibility of some PKC-based KMSs against battery exhaustion attack is investigated.

It is noteworthy that a few other approaches have been proposed for mitigating DOS attacks in PKC-based operations in WSN. For example, a puzzle- based approach was proposed in [35]. This scheme prevented DOS attack by using a weak authenticator that was verified efficiently but its forgery was computationally expensive for adversary. Another method was presented in [36] were PKC messages were dropped based on a probability to meet with the computation capability of nodes. While these two works try to address the same issue as our proposed methods, but the approach is quite different. More specifically, existing works only add some mechanisms for mitigating DOS attacks without proposing a new KMS. This become more evident when we see that their underlying KMS is the conventional certificate-based approach. On the other hand, our proposed methods replace certificate verification phase of PKC-based KMS with a novel symmetric-based approach that has much lower energy consumption.

The rest of this paper is organized as follows. Section 2 presents BA method and investigates its vulnerabilities to DOS attacks. Section 3 is devoted to the proposed methods. Security analysis and performance of the proposed methods are presented in section 4. Section 5 discusses the proposed methods and finally conclusions are drawn in section 6.

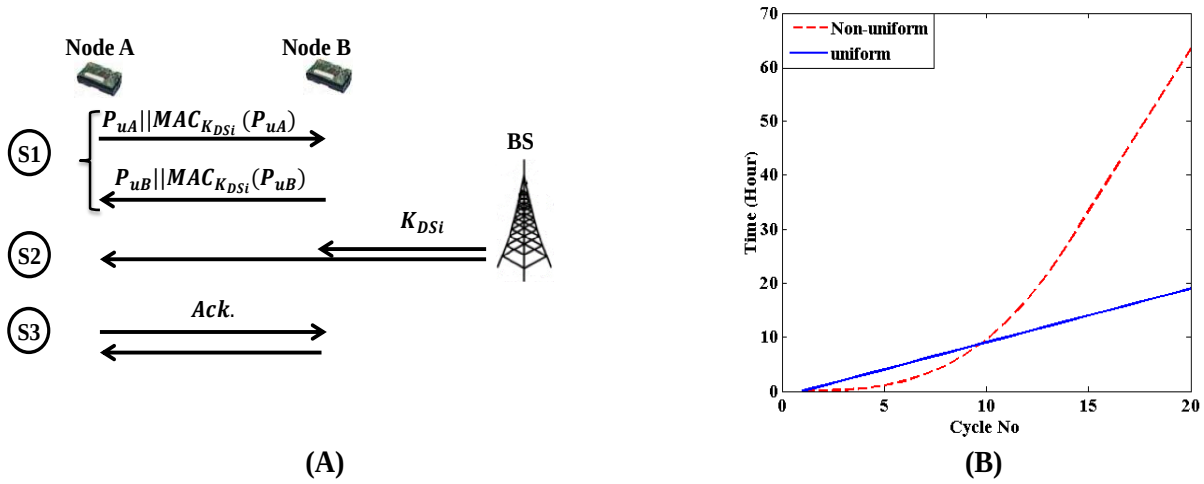


Figure 2. (A) Simple schematic of the basic method (B) Two possible trimming schedules [33]

II. BASIC METHOD AND ITS VULNERABILITY

The main idea of Broadcast-Authenticated (BA) method -which we will call it the basic method throughout this manuscript- is to replace the verification of digital certificates of PKC with a cheaper symmetric based mechanism. Details of basic method are as follows.

First, BS uses a hash function and generates $\{K_{DSi}\}$ key chain.

$$K_{DSn} \ldots \rightarrow K_{DS1} \rightarrow K_{DS0} \quad (1)$$

Then, for every node *x*, BS generates public (P_{ux}) and private (P_{rx}) parameters of elliptic curve Diffie-Hellman (ECDH) and then generates a set of one-time signatures for node *x* according to:

$$Sign_{x_i} = MAC_{K_{DSi}}(P_{ux}) \quad i = 1, \ldots, n \quad (2)$$

Nodes are then pre-loaded with their one-time signatures and their public and private ECDH parameters. Assuming that BS transmits K_{DSi} to nodes securely, two neighboring nodes can exchange their *i*th. one-time signatures and then use MAC method for authenticating public key of the other party. Finally, after generating the shared key according to Diffie-Hellman scheme, acknowledgment messages are exchanged. Fig. 2-A illustrates these steps.

Investigating fig. 2-A shows that, the second step of this method needs a mechanism to guarantee its freshness and authenticity; otherwise it would be vulnerable against replay and other attacks. To address this, the basic protocol employed a modified version of µTesla. To that end, BS generates µTesla key chain and preloads nodes with K_{Auth00}.

$$K_{Authn} \ldots \rightarrow K_{Auth1} \rightarrow K_{Auth0} \rightarrow K_{Auth00} \quad (3)$$

After network is deployed, BS uses a timing scheduling and broadcasts message 4

$$BS \longrightarrow X: E_{K_{Authi}}(K_{DSi}|| \, i|| \, \Delta_i) \quad (4)$$

where, i is the cycle number, and Δ_i is the locally computed time difference between current and the last time message 4 was broadcasted. After *t* seconds, BS broadcasts and reveals K_{Authi} for the nodes.

$$BS \longrightarrow X: K_{Authi} \quad (5)$$

Resiliency of this approach to different attacks was investigated throughly in [33] and interested reader may refer to it for further details.

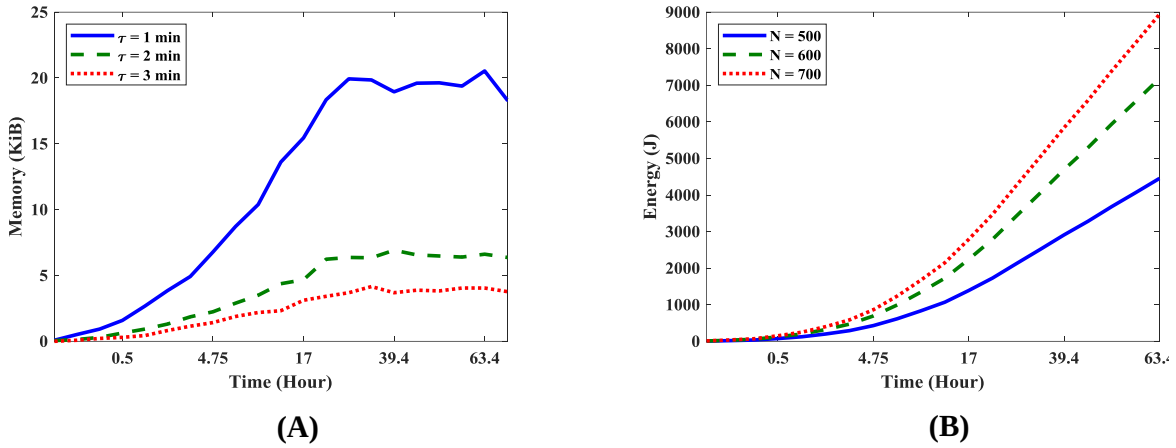
Each execution of this routine is called an authentication cycle and, authentication cycles are repeated according to a timing schedule. In the BA method the lifetime of the network was divided into a series of authentication-cycles, where all messages within the same cycle have to wait until the corresponding key is disclosed (message 5). Considering the duration of cycles, two scheduling approaches could be differentiated. In the uniform scheduling, the duration between consecutive cycles is kept constant, whereas, in the non-uniform case the duration is increased gradually [32]. The rationale behind non-uniform scheduling is that, after nodes are deployed, all of them should establish pairwise keys with their neighbors. But, as time goes by, fewer nodes need to participate in authentication cycles. Therefore, it is better to start with short cycles and then gradually increase duration of cycles. Fig. 2-B depicts these two timing schedules. It is noteworthy that all of analyses of this paper are based on non-uniform timing schedule.

Analysis and simulation results in [33] showed that basic method is very energy efficient. On the other hand, looking at the protocol we see that authentication of exchanged one-time signatures are delayed until the receipt of message 5. In the next section it is shown that this delayed authentication could lead to different DOS attacks.

A. *Vulnerabilities of the basic method*

1) *Network model:*

To investigate different scenarios, a series of simulations was conducted. To that end, different numbers of Mica2 nodes with transceiver range of 30 meters were uniformly distributed over a square field of 500×500 meters. Each simulation was run for 100 times and the final results were averaged. Furthermore, network used packet size of 41 bytes, 32 bytes for payload and 9 bytes for header [37].



(A) **(B)**

Figure 3. Flooding attacks (A) Exhausted memory of a node (B) Exhausted energy of network

2) Adversary model:

In the analysis, the adversary has limited computation power and he cannot break cryptographic primitives. His main interests are to inject false information, authenticate fake nodes to the network, and to mount DOS attacks to exhaust resources of network. To achieve these goals, he may eavesdrop communication between nearby nodes and later replay them. Also, he can compromise only limited number of nodes. Furthermore, he may have some agents dispersed throughout the network with exclusive communication channel for achieving more effective communications.

3) Flooding attack on the basic method

Delayed authentication, forces nodes to buffer messages and wait for BS to reveal the key. Adversary can take advantage of this delay and flood network with fake messages. If adversary can exhaust memory of nodes, the legitimate message may arrive when there is no memory left. To put this attack into perspective we conducted a simulation. We assumed that adversary picks his time of attack from a uniform distribution on the interval of [1,τ]. Fig. 3-A shows results of our analysis for different values of τ. It is quite clear, that as the value of τ decreases, nodes need larger memory to accommodate for buffering of received messages.

In addition to storing bogus messages, nodes will retransmit them for their neighbors [38]. This could lead to a severe energy exhaustion attack. To illustrate effect of this attack we conducted a simulation. we assumed that adversary picks his time of attack from a uniform distribution on the interval [0, 10] minutes. Then, the total amount of energy consumed by all nodes to re-broadcast those fake messages was computed. Fig. 3-B shows how this energy varies with time. It is quite clear, that as time goes on, more energy is exhausted from the network. Additionally, we see exhausted energy is positively correlated with the number of nodes in the network.

III. PROPOSED SCHEME

In this section we improve the basic method to alleviate its DOS vulnerabilities. We propose two different improved protocols that benefit from immediate authentication. Table I describes notations that are used in the rest of this paper.

TABLE I. NOTATIONS USED IN THIS PAPER

Notation	*Meaning*
||	Concatenation
P_{ux}	Public parameter of elliptic curve Diffie-Hellman of Node *x*
P_{rx}	Private parameter of elliptic curve Diffie-Hellman of Node *x*
i	Cycle number
K_{DSi}	Key used to generate *i*th. signature
Tx_i	Time measured locally at node *x*
$Sign_{xi}$	$Sign_{Xi} = MAC_{K_{DSi}}(Pu_X)$
$Ticket_{Xi}$	$Ticket_{Xi} = P_{uX} \| \| Sign_{Xi}$
Δ_i	Time difference between two consecutive cycles
K_{Authi}	µTesla key chain
$MAC_K(M)$	Message Authentication Code of message (*M*) using key (*K*)
$E_K(M)$	Symmetric encryption of message (*M*) using key (*K*)
$D_K(M)$	Symmetric decryption of message (*M*) using key (*K*)
K_{AB}	Pairwise key between node *A* and *B*
f, g	Some publicly agreed on functions
$h(M)$	Hash value of message *M*
γ	Maximum number of nodes that every node can authenticate in a single cycle

A. Broadcast Authenticated protocol with immediate authentication (i-BA)

In section 2, we saw that the delayed authentication of message (4) was the main source of DOS vulnerabilities. In order to solve such vulnerabilities, we need to analyze content of that message. Referring to message (4) we see that it is the encrypted version of a key (K_{DSi}), a counter (i), and a cycle duration (Δ_i). Investigating these components reveals that BS knows value of counter and both keys (K_{DSi}, K_{Authi}) for all cycles. Additionally, we know that BS manages timing of cycles; therefore, assuming that BS (at least) knows duration of next cycle (Δ_{i+1}) is quite logical.

Consequently, while initiating a new cycle, BS knows message of the next cycle. Based on these observations we propose a modified protocol with immediate authentication capabilities.

Assuming we are in the cycle *i*-1, BS will generate messages (4) for both current cycle and next cycle, that is BS generates $E_{K_{Authi-1}}(K_{DSi-1}||i-1||\Delta_{i-1})$ and $E_{K_{Authi}}(K_{DSi}||i||\Delta_i)$. Then instead of broadcasting message (4), message (6) is broadcasted.

$$BS \longrightarrow X: E_{K_{Authi-1}}(K_{DSi-1}||i-1||\Delta_{i-1})|| \\ E_{K_{Authi-1}}(h(E_{K_{Authi}}(K_{DSi}||\, i||\, \Delta_i))||i-1), \quad T_{BS} = T_{BSi-1} \tag{6}$$

which is the concatenation of message (4) for cycle *i*-1 and encrypted version of hash value of message (4) that is going to be sent in next cycle. Fig. 4 shows a schematic of this concept.

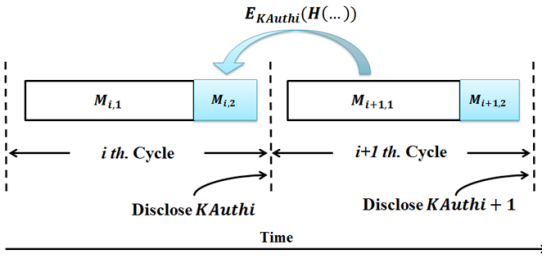


Figure 4. Protocol with Immediate Authentication

We now show nodes can authenticate message 6 immediately and there is no need to buffer it. Assume node *X* has received message (6) in the previous cycle, node *X* splits it into two parts,

$$\begin{cases} M_{i-1,1} = E_{K_{Authi-1}}(K_{DSi-1}||\, i-1||\, \Delta_i - 1) \\ M_{i-1,2} = E_{K_{Authi-1}}(h(E_{K_{Authi}}(K_{DSi}||i||\Delta_i)))||i-1) \end{cases} \tag{7}$$

node X waits until key $K_{Authi-1}$ is disclosed, and then decrypts $M_{i-1,2}$ and stores value of $h(E_{K_{Authi}}(K_{DSi}||\, i||\, \Delta_i))$ for the next cycle. For convenience we denote this message by μ_{i-1}. In the next cycle, X receives the following message:

$$BS \longrightarrow X: M_{i,1}||\, M_{i,2} \tag{8}$$

Fortunately, node X knows hash value of $M_{i,1}$ from the previous cycle and immediately checks its authenticity,

$$h(M_{i,1}) \stackrel{?}{=} \mu_{i-1} \tag{9}$$

Now, we present the whole *i*-BA protocol. Each authentication cycle starts with broadcasting of message (6) and nodes participating in that cycle could generate pair-wise keys with their neighbors. Assuming we are in cycle i (nodes can sync with network after decryption of $M_{i-1,1}$), upon broadcasting of message (6) BS stores its local time as T_{BSi}. Additionally, each node *X* stores its local time after receiving the message (6) as T_{Xi}. Let A and B denote two neighbouring nodes, first they check authenticity of message (6) using equation (9). Then, they generate a ticket by appending their public keys with their *i*th. one-time signatures $(Sign_{A_i}, Sign_{B_i})$ and then exchange their tickets. BS waits for some time *t* and then broadcast message (10)

$$BS \longrightarrow X: K_{Authi} \tag{10}$$

Because K_{Authi} constitutes a hash chain and nodes are preloaded with the last key, they can check authenticity of the received key. Then, nodes decrypt both $M_{i,1}$ and $M_{i,2}$ and compare value of received Δ_i with their locally calculated value, if those values were relatively the same, freshness of the messages is guaranteed, and they continue with the protocol and discard it otherwise. Then, nodes use K_{DSi} with MAC operation and check authenticity of the received tickets. After running ECDH and extracting their shared keys, node exchange acknowledgment messages. The complete *i*-BA protocol is presented in Table II.

B. *Bloom filter-based Broadcast Authenticated PKC (b-BA)*

Bloom filter is a data structure which supports membership queries very efficiently [39]. Bloom filter is an *m*-bit vector all initially set to 0. For representing the set $S = \{s_1, s_2, \ldots, s_n\}$, k independent hash functions are selected such that $h_i(M) \rightarrow [0, m-1]$, $1 \le i \le k$. Then, bits $h_i(s_j)$, $1 \le i \le k, 1 \le j \le n$ of this vector are set to 1. After bloom filter is constructed, for an element such as *x* if all bits

TABLE II. *i*-BA PROTOCOL

$M_{i,1} = E_{K_{Authi}}(K_{DSi}\|\|i\|\|\Delta_i), \quad M_{i,2} = E_{K_{Authi}}(h\left(E_{K_{Authi+1}}(K_{DSi+1}\|\|i+1\|\|\Delta_{i+1})\right)\|\|i)$ $T_1: \begin{cases} BS \rightarrow A: M_{i,1}\|\|M_{i,2} & , T_{BS} = T_{BSi},\ T_A = T_{Ai} \\ BS \rightarrow B: M_{i,1}\|\|M_{i,2} & , T_{BS} = T_{BSi},\ T_B = T_{Bi} \end{cases}$ $A: h(M_{i,1}) \stackrel{?}{=} \mu_{i-1}$, $\mu_{i-1}\|\|i-1 = D_{K_{Authi-1}}(M_{i-1,2}),\ \mu_{i-1} = h(E_{K_{Authi}}(K_{DSi}\|\|\, i\|\|\, \Delta_i))$ $B: h(M_{i,1}) \stackrel{?}{=} \mu_{i-1}$, $\mu_{i-1}\|\|i-1 = D_{K_{Authi-1}}(M_{i-1,2}),\ \mu_{i-1} = h(E_{K_{Authi}}(K_{DSi}\|\|\, i\|\|\, \Delta_i))$
$A: Ticket_{Ai} = [P_{uA}\|\|Sign_{Ai}], \quad Sign_{Ai} = MAC_{K_{DSi}}(Pu_A)$ $B: Ticket_{Bi} = [P_{uB}\|\|Sign_{Bi}], \quad Sign_{Bi} = MAC_{K_{DSi}}(Pu_B)$ $T_2: \begin{cases} A \rightarrow B: Ticket_{Ai} \\ B \rightarrow A: Ticket_{Bi} \end{cases}$
$T_3: \begin{cases} BS \rightarrow A: K_{Authi} & , T_{BS} = T_{BSi} + t,\ T_A \cong T_{Ai} + t \\ BS \rightarrow B: K_{Authi} & , T_{BS} = T_{BSi} + t,\ T_B \cong T_{Bi} + t \end{cases}$
A: $K_{Authi} \stackrel{h?}{\rightarrow} K_{Authi-1}$; $T_{Ai} - T_{Ai-1} \stackrel{?}{\cong} \Delta_i$; $Sign_{Bi} \stackrel{?}{=} MAC_{K_{DSi}}(P_{u_B})$ B: $K_{Authi} \stackrel{h?}{\rightarrow} K_{Authi-1}$; $T_{Bi} - T_{Bi-1} \stackrel{?}{\cong} \Delta_i$; $Sign_{Ai} \stackrel{?}{=} MAC_{K_{DSi}}(P_{u_A})$
$K_{AB} = f(P_{uB}.P_{rA}, i) = f(P_{uA}.P_{rB}, i)$
$X \rightarrow Y: g(K_{XY})$ $Y \rightarrow X: g(K_{XY} + 1)$

$h_i(x)$, $1 \le i \le k$ of bloom filter are equal to 1, that item *x* belongs to the set *S* with high probability.

Considering the knowledge of BS about timing schedule, let us go one step further and assume that BS uses a deterministic timing schedule. Therefore, BS knows all values of K_{DSi}, *i*, Δ_i and can construct the set $S=\{<K_{DS1}||1||\Delta_1>,<K_{DS2}||2||\Delta_2>,\ldots,<K_{DSL}||L||\Delta_L>\}$ where L is the maximum number of authentication cycles. Now, BS can use bloom filter for authenticating its messages. It is noteworthy that while method of Bloom-based authentication scheme (BAS) [30] is also based on bloom filter, but there are important differences between BAS and our proposed approach. BAS used bloom filter for authentication of public keys, whereas in our method broadcast messages are used for authentication of public keys. More specifically, *b*-BA method uses bloom filter for checking authenticity of broadcast messages from BS. Later on, in section 4, we show there are significant differences between performance of BAS and *b*-BA.

The proposed method works as follows; first, BS generates the signature key chain:

$$K_{DSL} \xrightarrow{h} \ldots \xrightarrow{h} K_{DS1} \xrightarrow{h} K_{DS0} \xrightarrow{h} K_{DS00} \quad (11)$$

After constructing the set S, BS constructs its bloom filter. Then, every node is preloaded with its public and private keys, its chain of signatures (2), the last key of key chain (K_{DS00}), and bloom filter of set S. After nodes are deployed, neighboring nodes exchange their tickets. Then, BS broadcasts message (12):

$$BS \rightarrow X: K_{DSi}||i||\Delta_i \quad (12)$$

Upon receiving this message, every node saves its local time (T_{Xi}) and checks authenticity of K_{DSi} by performing a hash function. Then integrity of message (12) is checked with bloom filter. Furthermore, freshness of message (12) is validated by comparing locally calculated time difference with the one sent from BS. If all of these security conditions are passed, nodes use K_{DSi} to check validity of received one-time signatures. Finally, nodes run ECDH and extracted shared keys. These steps are shown in Table III.

TABLE III. *b*-BA PROTOCOL

$A: Ticket_{Ai} = [P_{uA}\|\|Sign_{Ai}]$ $B: Ticket_{Bi} = [P_{uB}\|\|Sign_{Bi}]$ $T_1: \begin{cases} A \rightarrow B: Ticket_{Ai} \\ B \rightarrow A: Ticket_{Bi} \end{cases}$
$T_2: \begin{cases} BS \rightarrow A: K_{DSi}\|\|i\|\|\Delta_i & ,\Delta_i = (T_{BSi} - T_{BSi-1}) \\ BS \rightarrow B: K_{DSi}\|\|i\|\|\Delta_i & ,\Delta_i = (T_{BSi} - T_{BSi-1}) \end{cases}$
$A: \begin{cases} K_{DSi} \xrightarrow{h} K_{DSi-1}, Bloom(K_{DSi}\|\|i\|\|\Delta_i) \stackrel{?}{=} 1 \\ T_{Ai} - T_{Ai-1} \stackrel{?}{\cong} \Delta_i, \quad Sign_{Bi} \stackrel{?}{=} MAC_{K_{DSi}}(P_{u_B}) \end{cases}$ $B: \begin{cases} K_{DSi} \xrightarrow{h} K_{DSi-1}, Bloom(K_{DSi}\|\|i\|\|\Delta_i) \stackrel{?}{=} 1 \\ T_{Bi} - T_{Bi-1} \stackrel{?}{\cong} \Delta_i, \quad Sign_{Ai} \stackrel{?}{=} MAC_{K_{DSi}}(P_{u_A}) \end{cases}$
$K_{AB} = f(P_{uB}.P_{rA}, i) = f(P_{uA}.P_{rB}, i)$
$X \rightarrow Y: g(K_{XY})$ $Y \rightarrow X: g(K_{XY} + 1)$

IV. ANALYSIS OF THE PROPOSED METHODS

A. Security Analysis

1) Integrity

This service prevents the unauthorized alteration of data. In the *i*-BA protocol, adversary may try to modify message (6). This message consists of two distinct parts ($M_{i,1}$, $M_{i,2}$) both of which are encrypted with a key that is not disclosed yet. Therefore, their manipulation will produce a random message and it can be detected. $M_{i,2}$ consists of a counter, after decryption nodes use it to check integrity of $M_{i,2}$. Furthermore, this counter chains $M_{i,1}$ and $M_{i,2}$ together, thus preventing attacks like cut and paste [40]. In the *b*-BA protocol, integrity of (12) is the main concern. *b*-BA protocol employs two different mechanisms to provide security. In the first layer, bloom filter checks integrity of message (12). Furthermore, messages that pass bloom filter test should have certain values, otherwise they will be discarded. A through discussion on this subject is given in the subsection 5.

2) Authenticity

Authentication is usually divided into two categories. First, data origin authenticity provides assurance about source of message [27]. Both *i*-BA and *b*-BA protocols achieve this by using a hash based key chain. Second, entity authentication addresses identification of different parties of a protocol. Our methods provide this security service by means of one-time signatures of (2).

3) Freshness

To prevent the adversary from replaying old messages, nodes should be able to check freshness of messages. Message (6) in the *i*-BA and message (12) in the *b*-BA protocol are the main target of replay attacks.

a) i-BA protocol:

Considering the time of replaying message (6), three different scenarios are possible. First, replaying (6) in the same cycle and before disclosure of key K_{Authi}. Since message (6) is encrypted with K_{Authi} and this key is not disclosed yet, adversary neither can read its contents nor modify it. Therefore, in this scenario adversary only participates in the blind flooding and distributes message of BS to the nodes. In this scenario, network benefits from replay attack. Second, replaying (6) in another authentication cycle. In another cycle BS reveals K_{Authl}. Decryption of an old message (6) with K_{Authl} produces random bits (nodes can check this by looking at counter values), thus nodes discard such messages. Third, replaying (6) in the same authentication cycle but after K_{Authi} is disclosed. In this scenario, adversary can jam receiver of a target node to prevent it from getting message of BS. In this fashion adversary can use K_{Authi} and generate himself valid signature. Then, he can send the forged signature for the target node. Finally, he impersonates BS and discloses K_{Authi} for the target node.

According to table II, BS broadcasts message (6) and K_{Authi} at T_{BSi} and $T_{BSi}+t$ respectively. Consequently, adversary replays message (6) at $T_{Xi} > T_{BSi}+t$. According to table II, nodes locally calculate cycle duration and compare it with the one they receive:

$$T_{Ai} - T_{Ai-1} > \Delta_i + t \quad (13)$$

Equation (13) contradicts security condition of $T_{Ai} - T_{Ai-1} \stackrel{?}{\cong} \Delta_i$ thus this replay attack is detected. Adversary knows value of K_{Authi} and can try to change Δ_i to $\Delta_i + t$ in (6), but this act violates integrity of message (6). Later, we showed that the proposed method detects infringement of integrity, therefore, this attack is also detected.

b) b-BA protocol:

In the *b*-BA method two different scenarios are possible. First, replaying (12) in another cycle and second, replaying (12) in the same cycle. These scenarios coincide with the second and the third scenarios of the i-BA method, and they are detected.

4) Security and wormhole attack

In this attack two or more malicious nodes collaborate and set up a link with low latency [41]. This can be achieved by using powerful transmitters and another frequency band for exclusive communication. In this fashion node *X* gets messages (6) and (12) with negligible delay. Consequently, he can use disclosed key and generate valid signature without changing value of Δ_i.

Let us investigate authentication cycles. Cycle *i* starts with receiving of $K_{Authi-1}$ and K_{DSi-1} and lasts until K_{Authi} and K_{DSi} are received, in the *i*-BA and *b*-BA protocol respectively. If nodes terminate each authentication cycle *t* seconds before receiving of K_{Authi} and K_{DSi}, this attack is prevented. In this fashion when adversary generates his signatures, the cycle is terminated and thus nodes will discard his forged signatures.

5) False positive value of Bloom filter

Employing bloom filter introduces a false positive into the scheme. It means that bloom filter may suggest that an element *x* is in *S*, even though it is not. According to [42] the lowest value of this probability is equal to 2^{-k} and it is achieved for $k=(m.ln2)/n$, where *k*, *m*, and *n* represent the number of hash functions, length of bloom filter, and cardinality of set *S*. We want to estimate probability of forging message (12). Security of *b*-BA relies on two layers. In the first layer, message should pass bloom filter. A randomly generated message passes bloom filter with the probability of $2^{-ln2.(m/n)}$. In the second layer, contents of the message are checked. Message (12) consists of three components, a key (K_{DSi}), a cycle counter (*i*), and a time difference (Δ_i). Two of these components are deterministic and K_{DSi} should satisfy:

$$\underbrace{K_{DSi} \xrightarrow{h} \dots \xrightarrow{h} K_{DS0} \xrightarrow{h} K_{DS00}}_{i+1\ times} \quad (14)$$

Let L_k, L_i, and L_Δ denote length of K_{DSi}, *i*, and Δ_i in bits. Probability of a random message to pass second security layer is equal to:

$$\frac{2^{L_\Delta}}{2^{L_\Delta+L_i+L_k}} = 2^{-(L_i+L_k)} \quad (15)$$

Consequently, probability of forgery is equal to:

$$2^{-(L_i+L_k)} \times 2^{-ln2.(m/n)} \quad (16)$$

B. Connectivity of the proposed methods

Both methods assume that all nodes receive BS messages. Apparently, If BS is equipped with a powerful transmitter this assumption is correct. But if its range is limited, nodes can retransmit messages of BS for their neighbours. In this fashion BS messages can propagate through network.

Theorem1: Let p_{loss} and d denote probability of packet loss and number of neighbors of node *C*, then probability of *C* receiving message of BS satisfies:

$$p_r = 1 - {p_{loss}}^{d.p_r} \quad (17)$$

Proof: Among neighbors of node *C*, on average $d.p_r$ of them have received message of BS. Thus, probability of node *C* not receiving message of BS is:

$${p_{fail}} = {p_{loss}}^{d.p_r} \quad (18)$$

Theorem2: If two nodes participate in *l* authentication cycles, then probabilities of sharing a key in the *i*-BA and *b*-BA method are equal to:

$$P_{li-BA} = 1 - (1 - p_r^4)^l \quad (19)$$

$$P_{lb-BA} = 1 - (1 - p_r^2)^l \quad (20)$$

Proof: If nodes receive message of BS with probability of p_r and λ denotes the number of BS messages that node A should receive to run the protocol (λ is equal to 2 and 1 in the *i*-BA and *b*-BA methods), then probability of both nodes A and B receiving necessary messages of BS is:

$$P_{success} = p_r^\lambda . p_r^\lambda = p_r^{2\lambda} \quad (21)$$

If nodes A and B participate in *l* authentication cycles, probability of sharing a key would be equal to:

$$P_m = 1 - (1 - P_{success})^l \quad (22)$$

Fig. 5 shows how these probabilities change.

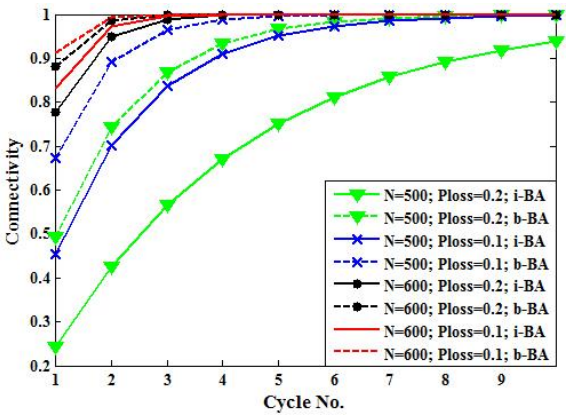


Figure 5. Probability of sharing a key after participating in *l* authentication cycles.

C. Energy consumption

To calculate energy consumption, cost of transmission [37] and executing cryptographic primitives [37, 43] were added together. Parameters of this calculation were as follows: $L_i = 10$ bits, $L_\wedge = 14$ bits, 128 bits for MAC and all keys, and 160 bits for ECDH keys. Furthermore, parameters of bloom filter were: $m=2^{15}$ bit, $k=23$, $n = 2^{L_i} = 1024$. Finally, SHA-1 and AES methods were used for symmetric

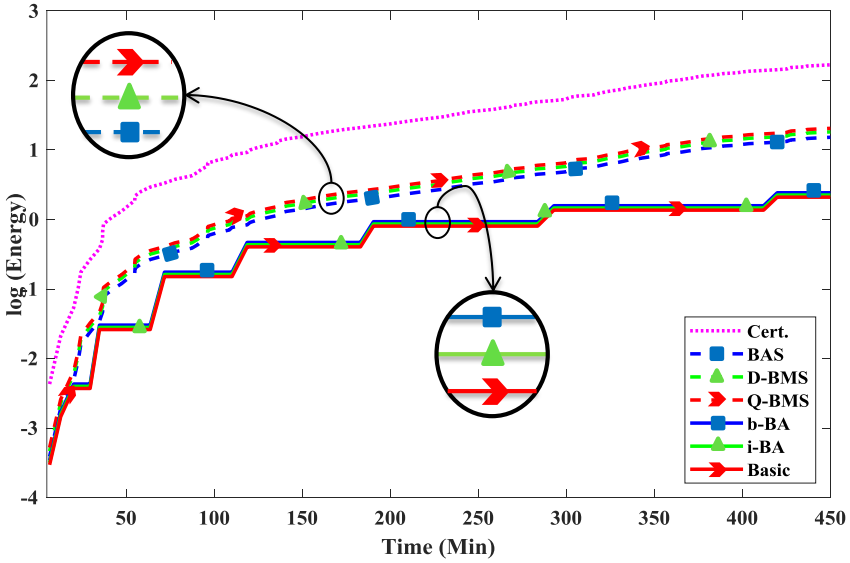


Figure 6. Exhausted energy of a single node

encryptions and hash operations. Adding communication and computation costs of table II and table III, total cost of *i*-BA and *b*-BA became 60.50 mj and 62.54 mj, respectively.

D. Resiliency against Battery Exhaustion Attack

Energy is the most precious resource in WSNs. Thus, numerous attacks have aimed to exhaust it. The main purpose of these attacks is to force victims to run costly operations. This section investigates resiliency of some PKC-based KMSs to these attacks. To that end, performances of the proposed methods were compared with certificate-based approach [37], Bloom-based authentication scheme (BAS) [30], Bloom-Merkle authentication scheme with double scalability (D-BMS) [30], and Bloom-Merkle authentication scheme with quadrupled scalability (Q-BMS) [30].

To mount an effective attack, adversary may listen to ongoing traffics ($Ticket_{X_i}$) and later re-transmits them for the victims. Such data will be authenticated and the whole protocol will be executed, therefore this scenario leads to the most severe case of battery exhaustion attack, and therefore was used for the analysis. We assumed that adversary picks his time of attack according to a uniform distribution on the interval of [0, 10] minutes. Furthermore, the maximum number of nodes that a node was allowed to authenticate in a single cycle was limited to 8 (γ=8). Fig. 6 shows exhausted energy of a single node for different PKC-based KMS in the logarithmic scale.

Referring to fig. 6, it is evident that all three versions of broadcast authenticated approach (basic, *i*-BA, *b*-BA) have the highest resiliency to battery exhaustion attack. Additionally, the performance of three versions are very similar. Therefore, adding the immediate authentication capability does not increase susceptibility of *i*-BA and *b*-BA to this attack.

E. Scalability

Let us assume that ID of nodes is 2 bytes and IDs of revoked nodes should be stored. Assuming 64KiB of memory, it is possible to calculate the maximum size of network in each method. *b*-BA uses 2^{15} bits of memory for storing bloom filter, therefore it supports (2^{19}-2^{15})/16=30720 nodes. On the other hand, *i*-BA method uses all this memory for revoked nodes therefore it can support up to 2^{19}/16=32678 nodes.

V. DISCUSSION

Nodes of WSNs have limited resources and hence are prone to different types of attacks. Thus, their protocols should address these problems. PKC-based KMS have many desirable characteristics. They provide perfect resiliency, perfect global, local, and node connectivity, and they have very good scalability and extensibility properties. But their energy consumption could be a heavy burden on tight resources of nodes. Therefore, reducing energy consumption of PKC-based KMS is very desirable. A possible solution is to replace checking of digital certificate with symmetric based methods. One such method relied on a set of one-time signatures. For this method to work, BS should convey some credentials to nodes securely. To that end, a modified version of µTesla was proposed in the basic method. According to fig. 3, this system is vulnerable to flooding attacks. Furthermore, these figures show that, effect of this DOS attack on the energy consumption is more severe.

To alleviate these problems, we assumed a more knowledgeable BS scenario. First, we assumed that BS knows exact time of the next authentication cycle. In this manner, hash value of the next message was appended to the current message. Second scheme went one step further and assumed that BS knows time of all authentication cycles. This knowledge enabled BS to use bloom filter for authentication purposes. Both of these assumptions added immediate authentication to the system and solved its vulnerability to flooding attacks.

While the proposed methods improved resiliency of KMS against DOS attacks, the proposed methods inherit other desirable characteristics of the basic method. One of those properties is addressing problem of dead nodes. Because battery of dead nodes has been depleted, they have lost their functionality. But they have very valuable information stored in them. Adversary can collect these nodes and exploit their

TABLE IV. Comparison between different PKC based key management systems

Scheme	*Energy cost*	*Scalability*	*Flooding Attack*	*Battery Exhaustion*	*Reference*
Certificate Based	187.6	Highest	Resilient	Very Low Resiliency	[37]
BAS	66.17	Very low	Resilient	Moderate Resiliency	[30]
D-BMS	71.51	low	Resilient	Moderate Resiliency	[30]
Q-BMS	75.26	Moderate	Resilient	Moderate Resiliency	[30]
Basic Method	58.68	Highest	Vulnerable	High Resiliency	[33]
i-BA	60.50	Highest	Resilient	High Resiliency	------
b-BA	62.54	Very High	Resilient	High Resiliency	------

information for mounting more effective attacks. For example, he can read their keying material and use them for programming his own nodes. Therefore, KMS should employ a reliable revocation mechanism. One solution is to broadcast ID of revoked nodes. Apparently, nodes have limited amount of memory and storing information of many dead nodes would be infeasible. Another solution that is very common to PKC-based system is to assign an expiration time to each certificate. In the proposed methods problem of dead nodes can effectively be addressed. The proposed methods rely on one-time signatures. Also, as BS reveals the corresponding key of a signature, that signature gets expired. Considering average lifetime of nodes, BS can preload nodes with suitable number of signatures such that when their battery is depleted, there would be no valid signature left. Consequently, dead nodes would not provide any useful information for adversary.

Comparing results of previous sections shows that both *i*-BA and *b*-BA have their own merits. *i*-BA consumes less energy and it supports larger networks, on the other hand in the *b*-BA scheme nodes share the common key much faster, especially when network in not dense or probability of packet loss is high (fig. 5). Furthermore, *b*-BA method is simpler, and its implementation would be easier. Finally, according to fig. 6, both of the proposed methods have good resiliency against battery exhaustion attack. Table IV presents a comparison between proposed methods and some previous PKC-based KMSs.

VI. CONCLUSION AND FUTURE WORK

There are many new challenges in WSNs due to different trade-offs and conflicting requirements. Tight constraints on resources -such as energy, memory and computation power- in addition to unique features of these networks, have turned most of algorithms from conventional networks impractical. Recently, security and key management system (KMS) in WSNs have received a lot of attention. Unfortunately, most of previous works have sacrificed security in favor of reducing the energy consumption. To avoid such trade-offs, one solution is to reduce energy consumption of PKC-based KMS. With this goal in mind, a recently proposed energy efficient KMS was analyzed in this paper. It was shown that delayed authentication of that method leads to some serious DOS attacks. Later, we showed that by extending knowledge of BS those vulnerabilities can be solved. To this end, two new methods based on adding hash value of the next message to the current message and bloom filter were proposed. Simulation results showed that these improved methods maintain energy efficiency, high scalability, and high resiliency against battery exhaustion attack of the original method. Furthermore, immediate authentication of these improved methods removed vulnerability of the basic method to flooding attacks.

For future works we would like to implement the proposed methods on WSN nodes and carry out further simulations and analysis.

REFERENCES

[1] Romer, K., & Mattern, F. (2004). The design space of wireless sensor networks. *Wireless Communications, IEEE, 11*(6), 54-61.

[2] Barnwal, R. P., Bharti, S., Misra, S., & Obaidat, M. S. (2014). UCGNet: wireless sensor network- based active aquifer contamination monitoring and control system for underground coal gasification. *International Journal of Communication Systems*.

[3] Kułakowski, P., Calle, E., & Marzo, J. L. (2013). Performance study of wireless sensor and actuator networks in forest fire scenarios. *International Journal of Communication Systems, 26*(4), 515-529.

[4] Sivaraman, V., Dhamdhere, A., Chen, H., Kurusingal, A., & Grover, S. (2013). An experimental study of wireless connectivity and routing in ad hoc sensor networks for real-time soccer player monitoring. *Ad Hoc Networks, 11*(3), 798-817.

[5] Jou, M., & Wang, J. (2013). Ubiquitous tutoring in laboratories based on wireless sensor networks. *Computers in Human Behavior, 29*(2), 439-444.

[6] Palafox, L. E., & García- Macías, J. A. (2009). Deploying a voice capture sensor network system for a secure ubiquitous home environment. *International Journal of Communication Systems, 22*(9), 1199-1212.

[7] Hamdi, M., Boudriga, N., & Obaidat, M. S. (2008). WHOMoVeS: An optimized broadband sensor network for military vehicle tracking. *International Journal of Communication Systems, 21*(3), 277-300.

[8] Eschenauer, L., & Gligor, V. D. (2002). *A key-management scheme for distributed sensor networks.* Paper presented at the Proceedings of the 9th ACM conference on Computer and communications security.

[9] Du, W., Deng, J., Han, Y. S., Varshney, P. K., Katz, J., & Khalili, A. (2005). A pairwise key predistribution scheme for wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC), 8*(2), 228-258.

[10] Liu, D., Ning, P., & Li, R. (2005). Establishing pairwise keys in distributed sensor networks. *ACM Transactions on Information and System Security (TISSEC), 8*(1), 41-77.

[11] Zhu, S., Setia, S., & Jajodia, S. (2006). LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks (TOSN), 2*(4), 500-528.

[12] Bechkit, W., Challal, Y., & Bouabdallah, A. (2013). A new class of Hash-Chain based key pre-distribution schemes for WSN. *Computer Communications, 36*(3), 243-255.

[13] Chan, H., Perrig, A., & Song, D. (2003). *Random key predistribution schemes for sensor networks.* Paper presented at the Security and Privacy, 2003. Proceedings. 2003 Symposium on.

[14] Camtepe, S. A., & Yener, B. (2007). Combinatorial design of key distribution mechanisms for wireless sensor networks. *Networking, IEEE/ACM Transactions on, 15*(2), 346-358.

[15] Bechkit, W., Challal, Y., Bouabdallah, A., & Tarokh, V. (2013). A highly scalable key pre-distribution scheme for wireless sensor networks. *IEEE Transactions on Wireless Communications*, *12*(2), 948-959.

[16] Das, A. K. (2015). A secure and effective biometric- based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. *International Journal of Communication Systems*.

[17] Wei, J., Hu, X., & Liu, W. (2014). Two- factor authentication scheme using attribute and password. *International Journal of Communication Systems*.

[18] Alcaraz, C., Lopez, J., Roman, R., & Chen, H.-H. (2012). Selecting key management schemes for WSN applications. *Computers & Security*, *31*(8), 956-966.

[19] Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. C. (2004). Comparing elliptic curve cryptography and RSA on 8-bit CPUs *Cryptographic Hardware and Embedded Systems-CHES 2004* (pp. 119-132): Springer.

[20] Misra, S., Goswami, S., Taneja, C., & Mukherjee, A. (2014). Design and implementation analysis of a public key infrastructure- enabled security framework for ZigBee sensor networks. *International Journal of Communication Systems*.

[21] Watro, R., Kong, D., Cuti, S.-f., Gardiner, C., Lynn, C., & Kruus, P. (2004). *TinyPK: securing sensor networks with public key technology*. Paper presented at the Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks.

[22] Das, M. L. (2009). Two-factor user authentication in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, *8*(3), 1086-1090.

[23] Yeh, H.-L., Chen, T.-H., Liu, P.-C., Kim, T.-H., & Wei, H.-W. (2011). A secured authentication protocol for wireless sensor networks using elliptic curves cryptography. *Sensors*, *11*(5), 4767-4779.

[24] Han, W. (2011). Weakness of a Secured Authentication Protocol for Wireless Sensor Networks Using Elliptic Curves Cryptography. *IACR Cryptology ePrint Archive*, *2011*, 293.

[25] Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography*: CRC press.

[26] Ren, K., Lou, W., Zeng, K., & Moran, P. J. (2007). On broadcast authentication in wireless sensor networks. *Wireless Communications, IEEE Transactions on*, *6*(11), 4136-4144.

[27] Cao, X., Kou, W., Dang, L., & Zhao, B. (2008). IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks. *Computer Communications*, *31*(4), 659-667.

[28] Lim, C. H. (2011). Secure code dissemination and remote image management using short-lived signatures in WSNs. *IEEE communications letters*, *15*(4), 362-364.

[29] Shim, K.-A., Lee, Y.-R., & Park, C.-M. (2013). EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks. *Ad Hoc Networks*, *11*(1), 182-189.

[30] Ren, K., Yu, S., Lou, W., & Zhang, Y. (2009). Multi-user broadcast authentication in wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, *58*(8), 4554-4564.

[31] Liu, Y., Li, J., & Guizani, M. (2012). PKC Based Broadcast Authentication using Signature Amortization for WSNs. *Wireless Communications, IEEE Transactions on*, *11*(6), 2106-2115.

[32] Ghasemzadeh, H., Aref, M. R., & Payandeh, A. (2013, August). A novel and low-energy PKC-based key agreement protocol for WSNs. In Information Security and Cryptology (ISCISC), 2013 10th International ISC Conference on (pp. 1-6). IEEE.

[33] Ghasemzadeh, H., Payandeh, A., & Aref, M. R. (2014b). Toward an Energy Efficient PKC-Based Key Management System for Wireless Sensor Networks. *The ISC International Journal of Information Security*, *6*(1), 53-66.

[34] Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., & Culler, D. E. (2002). SPINS: Security protocols for sensor networks. *Wireless networks*, 8(5), 521-534.

[35] Ning, P., Liu, A., & Du, W. (2008). Mitigating DoS attacks against broadcast authentication in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *4*(1), 1.

[36] Kim, D., & An, S. (2016). PKC-Based DoS attacks-resistant scheme in wireless sensor networks. *IEEE Sensors Journal*, *16*(8), 2217-2218.

[37] Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. C. (2005). *Energy analysis of public-key cryptography for wireless sensor networks*. Paper presented at the Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on.

[38] McCune, J. M., Shi, E., Perrig, A., & Reiter, M. K. (2005). *Detection of denial-of-message attacks on sensor network broadcasts*. Paper presented at the Security and Privacy, 2005 IEEE Symposium on.

[39] Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, *13*(7), 422-426.

[40] Wagner, D., & Schneier, B. (1996). *Analysis of the SSL 3.0 protocol*. Paper presented at the The Second USENIX Workshop on Electronic Commerce Proceedings.

[41] Hu, Y.-C., Perrig, A., & Johnson, D. B. (2003). *Packet leashes: a defense against wormhole attacks in wireless networks*. Paper presented at the INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies.

[42] Mitzenmacher, M. (2002). Compressed bloom filters. *IEEE/ACM Transactions on Networking (TON)*, *10*(5), 604-612.

[43] Potlapally, N. R., Ravi, S., Raghunathan, A., & Jha, N. K. (2003). *Analyzing the energy consumption of security protocols*. Paper presented at the Proceedings of the 2003 international symposium on Low power electronics and design.

Hamzeh Ghasemzadeh was born in Tehran, Iran, in 1984. He received the B.Sc. and M.Sc. degrees in telecommunications engineering. He is currently pursuing a dual Ph.D. degree in *communicative sciences and disorders* and *computational mathematics science and engineering* at Michigan State University, MI, USA. He was an Adjunct Professor with the Department of Electrical Engineering, Azad University, Damavand Branch, until 2016. He has been working on different aspects of audio signal processing, covering security driven applications of audio signals, audio forensics, and acoustic analysis of pathological impaired voices. He is currently involved in different projects including statistical signal processing and data mining of images acquired through high speed videoendoscopy from vocal folds, statistical modeling of connected speech in time and frequency domains, and development of state-of-the art laser projection high speed videoendoscopy system. His primary research interests are applying statistical signal processing and machine learning

techniques for solving different speech/voice related problems.

Ali Payandeh received the M.Sc. degree in Electrical Engineering from Tarbiat Modares University in 1994, and the Ph.D. degree in Electrical Engineering from K.N. Toosi University of Technology (Tehran, Iran) in 2006. From 1996 to 2006, he was a director of research at the Applied Science Research Association, Iran, where he was involved in research for secure satellite communications. He is now an assistant professor in the Department of Information and Communications Technology at Malek-e-Ashtar University of Technology, Iran. He has published more than 75 papers in international journals and conferences. His research interests include information theory, coding theory, cryptography, security protocols, secure communications, and satellite communications.

Mohammad Reza Aref received the B.Sc. degree in 1975 from the University of Tehran, Iran, and the M.Sc. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in Electrical Engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of Electrical Engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 290 technical papers in communications, information theory and cryptography in international journals and conferences proceedings. At the same time, during his academic activities, he has been involved in different political positions. First Vice President of I.R. Iran, Vice President of I.R. Iran and Head of Management and Planing Organization, Minister of ICT of I.R. Iran and Chancellor of University of Tehran, are the most recent ones. His current research interests include areas of communication theory, information theory, and cryptography.